RECEIVED
CENTRAL FAX CENTER
DEC 21 2004

FAX TRANSMITTAL COVER SHEET

CONLEY ROSE, P.C.
600 Travis, Suite 7100
Houston, Texas 77002
Fax Number: (713) 238-8008
Telephone Number: (713) 238-8000

ORIGINAL WILL FOLLOW VIA:
- ☐ MAIL
- ☐ INTERNATIONAL AIRMAIL
- ☐ COURIER
- ☒ WILL NOT FOLLOW
- ☐ HAND DELIVERY
- ☐ WITH ENCLOSURE(S)
- ☐ WITHOUT ENCLOSURE(S)

PLEASE DELIVER THE FOLLOWING PAGES IMMEDIATELY TO:

NAME: **EXAMINER A. M. LEZAK, GROUP ART UNIT 2143**

FIRM: **U.S. PATENT & TRADEMARK OFFICE**

CITY: **ALEXANDRIA, VA**

FAX NO: **(703) 872-9306**

REMARKS: **Serial No. 09/652,834, filed 08/31/2000.**
Attached hereto is an Appeal Brief for filing with the U.S. Patent and Trademark Office. Please acknowledge receipt of this facsimile.

Total Number of Pages (Including This One): **THIRTY (30)**

FROM: Jonathan M. Harris, Direct Dial No. 713/238-8045

DATE: December 21, 2004

CLIENT/MATTER NO. 200301786-1 (1662-27800)

IF YOU DO NOT RECEIVE ALL THE PAGES,
PLEASE CALL BACK AS SOON AS POSSIBLE.

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

ORIGINAL

PATENT APPLICATION

ATTORNEY DOCKET NO. 200301786-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Michael S. BERTONE et al.	**Confirmation No.:**	4372
Application No.:	09/652,834	**Examiner:**	A. M. Lezak
Filing Date:	08/31/2000	**Group Art Unit:**	2143
Title:	SPECULATIVE DIRECTORY WRITES IN A DIRECTORY BASED CACHE COHERENT NONUNIFORM MEMORY ACCESS PROTOCOL		

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 11/09/2004.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$500.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

( ) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit:_____

OR

(X) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (703) 872-9306 on 12/21/2004

Number of pages: 29

Typed Name: Christina L. Paz

Signature:

Respectfully submitted,

Michael S. BERTONE et al.

By:

Jonathan M. Harris

Attorney/Agent for Applicant(s)
Reg. No. 44,144

Date: 12/21/2004

Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

COPY

PATENT APPLICATION

ATTORNEY DOCKET NO. 200301786-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Michael S. BERTONE et al.	**Confirmation No.:**	4372
Application No.:	09/652,834	**Examiner:**	A. M. Lezak
Filing Date:	08/31/2000	**Group Art Unit:**	2143
Title:	SPECULATIVE DIRECTORY WRITES IN A DIRECTORY BASED CACHE COHERENT NONUNIFORM MEMORY ACCESS PROTOCOL		

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 11/09/2004.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$500.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

( ) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit:_____

OR

(X) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (703) 872-9306 on 12/21/2004

Number of pages: 29

Typed Name: Christina L. Paz

Signature: Christina Paz

Respectfully submitted,

Michael S. BERTONE et al.

By: Jonathan M. Harris

Jonathan M. Harris
Attorney/Agent for Applicant(s)
Reg. No. 44,144

Date: 12/21/2004

Telephone No.: (713) 238-8000

RECEIVED
CENTRAL FAX CENTER

DEC 2 1 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Michael S. Bertone et al.	§	Confirmation No.:	4372
Serial No.:	09/652,834	§	Group Art Unit:	2143
Filed:	08/31/2000	§	Examiner:	A. M. Lezak
For:	Speculative Directory Writes In A Directory Based Cache Coherent Nonuniform Memory Access Protocol	§	Docket No.:	200301786-1

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: December 21, 2004

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed via facsimile on November 9, 2004.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas, through its merger with Compaq Computer Corporation (CCC) which owned Compaq Information Technologies Group, L.P. (CITG). The assignment from the CCC to CITG was recorded on January 15, 2002, at Reel/Frame 012478/0225. The Change of Name document was recorded on December 2, 2003, at Reel/Frame 014177/0428.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

Originally filed claims:	1-10.
Claim cancellations:	None.
Added claims:	11-16.
Presently pending claims:	1-16.
Presently appealed claims:	1-16.

IV. STATUS OF THE AMENDMENTS

No claims were amended after the final Office action dated September 13, 2004.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The summary is set forth in the following exemplary embodiments that correspond to claims involved in the appeal. Discussions about elements and recitations of these claims can be found at least at the cited locations in the specification and drawings.

In accordance with one embodiment, a distributed multiprocessing computer system (e.g., Figure 1) comprises a plurality of processor nodes (processors 100), a Home processor node (e.g., nodes 520 and 526 in Figures 5a and 5b, respectively), an Owner processor node (e.g., nodes 522 and 528 in Figures 5a and 5b, respectively), and a Requestor processor node (e.g., nodes 518 and 524 in Figures 5a and 5b, respectively). Each processor node, including the Home and Owner nodes, is coupled to an associated memory module (e.g., memory 192 in Figure 1) and each memory module may store data that is shared between the processor nodes. The Home processor node includes a data block and a coherence directory for the data block in its associated memory module. See e.g. Applicants' disclosure, page 27. The Owner processor node includes a copy of the data block in its associated memory module. See e.g. Applicants' disclosure, page 28. Further, the Requestor processor node encounters a read or write miss of the data block and requests the data block from the Home processor node. The Home processor node receives the request for the data block from the Requestor processor node, forwards the request to the Owner processor node for the data block and performs a speculative write of the next directory state to the coherence directory for the data block without waiting for the Owner processor node to respond to the request. See e.g. Applicants' disclosure, pages 28-32.

In accordance with another embodiment, a method comprises receiving a request for a data block, forwarding the request to an owner node (e.g., nodes 522 and 528 in Figures 5a and 5b, respectively) at which an updateable directory state of the data block is stored, and speculatively writing the directory state before receiving a coherence response from the owner. See e.g. Applicants' disclosure, pages 28-32.

In accordance with another embodiment, an apparatus is adapted to communicate with an owner node that is configured to have an exclusive copy of a data block. In this embodiment, the apparatus comprises memory (e.g., memory 102 in Figure 1) in which a directory table is stored, wherein the directory table includes a configurable cache state associated with the data block. See Applicants' disclosure, pages 25-26. The apparatus also comprises a cache controller (e.g., controller 326 in Figure 3; see also Applicants' disclosure, page 22) that speculatively updates the data block's cache state in the directory table upon receiving a memory request and before the apparatus receives a coherence response from the owner node. See e.g. Applicants' disclosure, pages 28-32.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-16 are obvious under 35 U.S.C. § 103 over Cherabuddi (U.S. Pat. No. 6,496,917) in view of Arimilli (U.S. Pat. No. 5,895,484).

VII. ARGUMENT

A. Overview of Cherabuddi

Cherabuddi is directed to a method to reduce latency in the event of a "cache miss." A cache miss occurs when data requested, for example, in a read transaction is not in the cache (a "miss") and the requested data must be retrieved from another location such as main memory. Cherabuddi addresses this issue as follows:

> In accordance with the present invention, when there is a cache miss in a CPU, the cache controller routes an address request for primary memory directly to the primary memory via the CPU as a speculative request, and also issues the address request to the system bus to maintain data coherency. The speculative request is queued in the primary memory controller, and thereafter retrieves speculative data from a specified primary memory address. The CPU monitors the system bus for requests to the specified primary memory address. If a subsequent transaction requesting the specified data is the read request that was issued on the system bus in response to the cache miss, the speculative request and any data retrieved thereby is validated and becomes non-speculative. If, on the other hand, the subsequent transaction requesting the specified data is a write transaction, the speculative request is canceled. The write transaction for the specified data is then processed before the read transaction for the specified data in order to maintain data coherency.

Column 3, lines 5-23.

Cherabuddi thus discloses speculatively reading data from memory associated with a central processing unit (CPU) and, at the same time, issuing a read request for the same data to the other processors in the system. Based on whether the requested data is located at another processor (as determined from monitoring the system bus for subsequent reads or writes of the data issued by the other processors), the CPU initiating the read transaction validates or cancels its speculative read of the data from its own memory. If the data was, in fact, located in the originating CPU's own memory, then the speculative read of that data is permitted to complete. However, if the data turns out to be located in other CPU's memory, then the speculative read is canceled. Cherabuddi does

not discuss cache directories, reading or writing from cache directories, or even speculatively reading/writing cache directories.

B. Overview of Arimilli

Arimilli is directed to a method for speculatively accessing cache memory. See Title. Arimilli solves the problem of a "retry" mechanism interrupting a read transaction. "If one processing unit responds with an intervention while another processing unit responds with a "retry," under a well-established rule, the retry response automatically overrules the intervention response. As a result, if there is an outstanding retry request by any one of the processing units on the system bus, the processing unit that contains the data will not issue a data bus request." Col. 1, lines 51-57. Arimilli defines an "intervention" as a "transfer of data from one processing unit to another processing unit on a system bus without going through a system memory." Col. 1, lines 31-33.

Figure 3 exemplifies Arimilli's solution. Figure 3 shows a "logic flow diagram of a method for speculatively sourcing data among cache memories." Col. 4, lines 61-62. As shown, Arimilli's method includes beginning to source data (block 39) from a source location before it is determined whether a retry will occur. If a retry does occur, then in block 42, the speculative act of data sourcing is canceled; otherwise, the sourcing is permitted to complete followed by an update in block 43 to the directory state. Col. 5, lines 29-53. In Arimilli, what is being done speculatively is reading data from memory, not writing data. Further, Arimilli teaches updating the directory after it is determined that a retry is not present and after the speculative data sourcing is completed. As such, the directory update in Arimilli is not performed speculatively.

C. Claims 1, 3, 5, 7, and 10

Appellants select claim 1 as representative of this claim group. Claim 1 requires, among other limitations, that "said Home processor node receives the request for the data block from the Requestor processor node, forwards the request to the Owner processor node for the data block and performs a speculative write of the next directory state to the coherence directory for the data block without waiting for the Owner processor node to respond to the request."

Claim 1 thus requires speculatively writing the next directory state of a coherence directory.

The above-quoted limitation is not disclosed in either Cherabuddi or Arimilli. Neither reference teaches speculatively writing anything and certainly not speculatively writing a next directory state. Cherabuddi teaches speculatively reading data from memory while concurrently issuing a read request to other processors for the same data. Arimilli teaches speculatively sourcing data and subsequently, that is not speculatively, updating a cache directory.

Based on the foregoing, Appellants respectfully submit that the Examiner erred in rejecting the claims in this group and that the Examiner's rejections should be reversed.

D. Claim 11

Claim 11 is a method claim that requires "speculatively writing the directory state before receiving a coherence response from the owner." In rejecting claim 11, the Examiner simply stated that "all limitations are addressed relative to claims 1, 5 and 10 above." As explained above, however, none of the art of record teaches or even suggests speculatively writing anything, much less a directory state. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 11 and that the Examiner's rejection should be reversed.

E. Claim 14

Claim 14 is directed to an apparatus that comprises a "cache controller that speculatively updates the data block's cache state in the directory table upon receiving a memory request and before the apparatus receives a coherence response from the owner node." In rejecting claim 14, the Examiner simply stated that "all limitations are addressed relative to claims 1, 5 and 10 above." As explained above, however, none of the art of record teaches or even suggests a cache controller that speculatively writes a directory state before receiving a coherence response from the owner. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 14 and that the Examiner's rejection should be reversed.

F. Claims 2, 6

Claims 2 and 6 depend from claims 1 and 5, respectively. At least for the reasons argued above with regard to claims 1 and 5, Appellants believe the Examiner erred in rejecting claims 2 and 6.

Further, claim 2 (which Appellants select to be representative of this claim group) requires the speculative write of the next directory state to occur "only if the next directory state cannot be determined and the Home processor node and Owner processor node are two different processor chips in the computer system." For these limitations, the Examiner specifically identified col. 3, lines 1-34 of Cherabuddi. This passage is as follows:

> A multiprocessor system includes a plurality of central processing units (CPUs) connected to one another by a system bus. Each CPU includes a cache controller to communicate with its cache, and a primary memory controller to communicate with its primary memory. In accordance with the present invention, when there is a cache miss in a CPU, the cache controller routes an address request for primary memory directly to the primary memory via the CPU as a speculative request, and also issues the address request to the system bus to maintain data coherency. The speculative request is queued in the primary memory controller, and thereafter retrieves speculative data from a specified primary memory address. The CPU monitors the system bus for requests to the specified primary memory address. If a subsequent transaction requesting the specified data is the read request that was issued on the system bus in response to the cache miss, the speculative request and any data retrieved thereby is validated and becomes non-speculative. If, on the other hand, the subsequent transaction requesting the specified data is a write transaction, the speculative request is canceled. The write transaction for the specified data is then processed before the read transaction for the specified data in order to maintain data coherency.
>
> Thus, in contrast to prior art architectures in which access to primary memory commences only after data coherency is established on the system bus, present embodiments, in response to a cache miss, route a speculative address request directly from a CPU's cache controller to its primary memory without arbitrating access to the system bus. By accessing primary memory contemporaneously with system bus arbitration, as opposed to sequentially arbitrating system bus access and then accessing primary memory, present

> embodiments may reduce primary memory latencies, which in turn improves CPU performance.

Appellants find absolutely no teaching or suggestion in this passage of the limitations of claim 2. There is no teaching of a speculative or, for that matter, a non-speculative write of a directory state. Further, there is no teaching of a speculative directory write under the condition specified which is "only if the next directory state cannot be determined and the Home processor node and Owner processor node are two different processor chips in the computer system." Arimilli is also deficient in this regard.

Based on the foregoing, Appellants respectfully submit that the Examiner erred in rejecting the claims in this group and that the Examiner's rejections should be reversed.

G. Claims 4, 8

Claims 4 and 8 depend from claims 1 and 5, respectively. At least based on the reasons argued above with regard to claims 1 and 5, Appellants believe the Examiner erred in rejecting claims 4 and 8.

Further, claim 4 (which Appellants select to be representative of this claim group) requires "correcting the next directory state for the data block if the response by the Owner processor node to the Home processor node request for the data block indicates a different next directory state from the next directory state speculatively written by the Home processor node to the coherence directory for the data block." For these limitations, the Examiner specifically identified col. 3, lines 1-35 and col. 4, lines 25-61 of Cherabuddi. These passages are as follows (the passage of col. 3, lines 1-35 is repeated below for convenience):

> A multiprocessor system includes a plurality of central processing units (CPUs) connected to one another by a system bus. Each CPU includes a cache controller to communicate with its cache, and a primary memory controller to communicate with its primary memory. In accordance with the present invention, when there is a cache miss in a CPU, the cache controller routes an address request for primary memory directly to the primary memory via the CPU as a speculative request, and also issues the address request to the system bus to

maintain data coherency. The speculative request is queued in the primary memory controller, and thereafter retrieves speculative data from a specified primary memory address. The CPU monitors the system bus for requests to the specified primary memory address. If a subsequent transaction requesting the specified data is the read request that was issued on the system bus in response to the cache miss, the speculative request and any data retrieved thereby is validated and becomes non-speculative. If, on the other hand, the subsequent transaction requesting the specified data is a write transaction, the speculative request is canceled. The write transaction for the specified data is then processed before the read transaction for the specified data in order to maintain data coherency.

Thus, in contrast to prior art architectures in which access to primary memory commences only after data coherency is established on the system bus, present embodiments, in response to a cache miss, route a speculative address request directly from a CPU's cache controller to its primary memory without arbitrating access to the system bus. By accessing primary memory contemporaneously with system bus arbitration, as opposed to sequentially arbitrating system bus access and then accessing primary memory, present embodiments may reduce primary memory latencies, which in turn improves CPU performance.

In accordance with the present invention, the multiprocessor computer system 100 shown in FIGS. 4 and 5 reduces primary memory latencies over prior art multiprocessor architectures by routing speculative address requests directly from the external cache controller 110 to the memory interface unit 120 via the CPU 101 in response to external cache misses. The memory interface unit 120 may begin processing speculative address requests before data coherency information is available from the system bus 11.

In accordance with the present invention, the speculative address requests received directly from the external cache controller 110 are reconciled with data coherency information provided by the system bus 11. Data coherency is maintained on the system bus 11 by issuing all address requests from the external cache controller lie to the system bus 11. Thus, for instance, if a write transaction for a specified address precedes a read transaction to the specified address in the execution order of the computer program, the write transaction is issued to the system bus 11 before the read transaction is issued to the system bus 11, irrespective of whether the two transactions are executed by the same CPU 101 or by different CPUs 101. Thus, the CPUs 101 monitor the system bus 11

> for requests corresponding to speculative address requests being processed in respective memory interface units 120. If the CPU 101 receives from the system bus 11 a write request to an address specified by a speculative request, the speculative address request, and any data retrieved thereby, are canceled. If, on the other hand, the CPU 101 receives from the system bus 11 an address request that corresponds to the speculative request before receiving a write-back transaction for the specified address, the speculative request is validated and becomes a non-speculative address request. If speculative data has already been retrieved from primary memory 25, the speculative data is validated, and thereafter processed by the CPU 101 in a conventional manner.

Appellants find absolutely no teaching or suggestion in these passages of the limitations of claim 4. There is no teaching of correcting the next directory state nor correcting the next directory state under the stated condition which is "if the response by the Owner processor node to the Home processor node request for the data block indicates a different next directory state from the next directory state speculatively written by the Home processor node to the coherence directory for the data block." Arimilli is also deficient in this regard.

Based on the foregoing, Appellants respectfully submit that the Examiner erred in rejecting the claims in this group and that the Examiner's rejections should be reversed.

H. Claim 9

The Examiner erred in rejecting claim 9 at least for the same reason as its independent claim (claim 5). Further, the Examiner erred in rejecting claim 9 for additional reasons. Claim 9 requires that the "speculative write of the next directory state releases hardware contained in the first processor node, allowing said first processor node to accept requests for data blocks and coherency directories for said data blocks stored in the memory module of the first processor node." For these limitations, the Examiner turned to Cherabuddi col. 4, line 62 through col. 5, line 6 which is as follows:

> The CPUs 101 also monitor the system bus 11 for snoop information provided by the other CPUs. If the snoop information indicates that specified data in primary memory is stale, all corresponding address

> requests to primary memory, and any data retrieved thereby, are canceled. The CPU that returns the dirty snoop information then routes updated data to the appropriate CPU 101 via the system bus 11. Conversely, if the snoop information indicates that specified data is clean, the corresponding address request is completed, and data retrieved thereby is thereafter processed in a conventional manner. When this condition occurs, primary memory latency is reduced over prior art multiprocessor computer architectures.

Appellants find no teaching or suggestion in the above-quoted passage of a speculative directory state write causing the release of hardware. Arimilli is also deficient in this regard. Based on the foregoing, Appellants respectfully submit that the Examiner erred in rejecting claim 9 and that the Examiner's rejection should be reversed.

I. Claim 12

The Examiner erred in rejecting claim 12 at least for the same reason as its independent claim (claim 11). Further, the Examiner erred in rejecting claim 12 for an additional reason. Claim 12 requires "speculatively writing the directory state before receiving a coherence response from the owner" (from independent claim 11) and "rewriting the directory state upon receipt of a coherency response from the owner" (added limitation per claim 12). Appellants find no teaching or suggestion in any of the art of record for this combination of limitations. In rejecting claim 12, the Examiner simply pointed to the rejections of claim 4 and 8. As should be apparent from Appellants' argument above regarding claim 4, the limitations of claim 12 are not disclosed in the art of record. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 12 and that the Examiner's rejection should be reversed.

J. Claim 13

The Examiner erred in rejecting claim 13 at least for the same reason as its independent claim (claim 11). Further, the Examiner erred in rejecting claim 13 for an additional reason. Claim 13 requires "confirming the speculatively written directory state upon receipt of a coherency response from the owner." Appellants find no teaching or suggestion in any of the art of record for this limitation.

When rejecting claim 13, the Examiner simply pointed to the rejections of claims 4 and 8. Neither of those claims requires confirming the speculatively written cache state as recited in claim 13. As such, the Examiner's rejection of claim 13 is unsupported. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 13 and that the Examiner's rejection should be reversed.

K. Claim 15

The Examiner erred in rejecting claim 15 at least for the same reason as its independent claim (claim 14). Further, the Examiner erred in rejecting claim 15 for an additional reason. Claim 15 requires that, "upon receiving the coherence response from the owner node, the cache controller confirms the speculatively updated cache state if the updated cache state comports with the cache information provided in the coherence response." Appellants find no teaching or suggestion in any of the art of record for this limitation. Appellants find no recitation in the art of record of speculatively updating a cache state or a cache controller that confirms a speculatively updated cache state if the updated state comports with certain cache information.

When rejecting claim 15, the Examiner simply pointed to the rejections of claims 4 and 8. Neither of those claims requires confirming the speculatively updated cache state as recited in claim 15. As such, the Examiner's rejection of claim 15 is unsupported. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 15 and that the Examiner's rejection should be reversed.

L. Claim 16

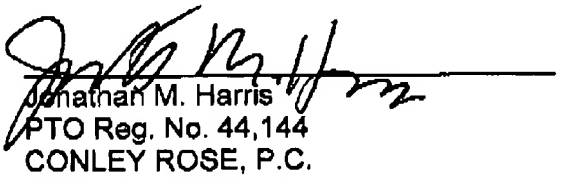
The Examiner erred in rejecting claim 16 at least for the same reason as its independent claim (claim 14). Further, the Examiner erred in rejecting claim 16 for an additional reason. Claim 16 requires that, "upon receiving the coherence response from the owner node, the cache controller changes the speculatively updated cache state if the updated cache state does not comport with the cache information provided in the coherence response." Appellants find no teaching or suggestion in any of the art of record for this limitation. Appellants

find no recitation in the art of record of speculatively updating a cache state or a cache controller that changes a speculatively updated cache state if the updated state does not comport with certain cache information. The Examiner simply referred to the rejection of claims 4 and 8 when rejecting claim 16. Based on Appellants' arguments above regarding claim 4, it should be apparent that the art of record does not disclose the limitations of claim 16. Accordingly, Appellants respectfully submit that the Examiner erred in rejecting claim 16 and that the Examiner's rejection should be reversed.

VIII. CONCLUSION

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,



Jonathan M. Harris
PTO Reg. No. 44,144
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

IX. CLAIMS APPENDIX

1. (Original) A distributed multiprocessing computer system, comprising:

a plurality of processor nodes each coupled to an associated memory module, wherein each memory module may store data that is shared between said processor nodes;

a Home processor node that includes a data block and a coherence directory for said data block in an associated memory module;

an Owner processor node that includes a copy of said data block in a memory module associated with the Owner processor node, said copy of said data block residing exclusively in said memory module;

a Requestor processor node that encounters a read or write miss of said data block and requests said data block from the Home processor node; and

wherein said Home processor node receives the request for the data block from the Requestor processor node, forwards the request to the Owner processor node for the data block and performs a speculative write of the next directory state to the coherence directory for the data block without waiting for the Owner processor node to respond to the request.

2. (Original) The distributed multiprocessing computer system of claim 1, wherein the speculative write of the next directory state occurs only if the next directory state cannot be determined and the Home processor node and Owner processor node are two different processor chips in the computer system.

3. (Original) The distributed multiprocessing computer system of claim 1, wherein the memory module containing the coherence directory for the data block is in a low latency state that reduces memory read and write access times while the Home processor node is performing the speculative write of the next directory state to the coherence directory for the data block.

4. (Original) The distributed multiprocessing computer system of claim 1, further comprising correcting the next directory state for the data block if the response by the Owner processor node to the Home processor node request for the data block indicates a different next directory state from the next directory state speculatively written by the Home processor node to the coherence directory for the data block.

5. (Original) A distributed multiprocessing computer system, comprising:

a plurality of processor nodes each coupled to an associated memory module, wherein each memory module may store data that is shared between said processor nodes;

a first processor node that includes a data block and a coherence directory for said data block in an associated memory module;

a second processor node that includes a copy of said data block in a memory module associated with the second processor node, said copy of said data block residing exclusively in said memory module;

a third processor node that encounters a read or write miss of said data block and requests said data block from the first processor node; and

wherein said first processor node receives the request for the data block from the third processor node, forwards the request to the second processor node for the data block and performs a speculative write of the next directory state to the coherence directory for the data block without waiting for the second processor node to respond to the request, said next directory state selected to reduce read-modify-write sequences.

6. (Original) The distributed multiprocessing computer system of claim 5, wherein the speculative write of the next directory state occurs only if the next directory state cannot be determined and the first processor node and second processor node are two different processor chips in the computer system.

7. (Original) The distributed multiprocessing computer system of claim 5, wherein the memory module containing the coherence directory for the data block is in a low latency state that reduces memory read and write access times while the first processor node is performing the speculative write of the next directory state to the coherence directory for the data block.

8. (Original) The distributed multiprocessing computer system of claim 5, further comprising correcting the next directory state for the data block if the response by the second processor node to the first processor node request for the data block indicates a different next directory state from the next directory state speculatively written by the first processor node to the coherence directory for the data block.

9. (Original) The distributed multiprocessing computer system of claim 5, wherein the speculative write of the next directory state releases hardware contained in the first processor node, allowing said first processor node to accept requests for data blocks and coherency directories for said data blocks stored in the memory module of the first processor node.

10. (Original) A distributed multiprocessing computer system, comprising:

a plurality of processor nodes each coupled to an associated memory module, wherein each memory module may store data that is shared between said processor nodes;

a Home processor node that includes a data block and a coherence directory for said data block in an associated memory module;

an Owner processor node that includes a copy of said data block in a memory module associated with the Owner processor node, said copy of said data block residing exclusively in said memory module;

a Requestor processor node that encounters a read or write miss of said data block and requests said data block from the Home processor node;

wherein said Home processor node receives the request for the data block from the Requestor processor node, forwards the request to the Owner processor node for the data block and performs a speculative write of the next directory state to the coherence directory for the data block without waiting for the Owner processor node to respond to the request; and

a disk drive coupled to each of said plurality of processor nodes.

11. (Previously presented) A method, comprising:

receiving a request for a data block;

forwarding the request to an owner node at which an updateable directory state of the data block is stored; and

speculatively writing the directory state before receiving a coherence response from the owner.

12. (Previously presented) The method of claim 11 further comprising re-writing the directory state upon receipt of a coherency response from the owner.

13. (Previously presented) The method of claim 11 further comprising confirming the speculatively written directory state upon receipt of a coherency response from the owner.

14. (Previously presented) An apparatus adapted to communicate with an owner node that is configured to have an exclusive copy of a data block, the apparatus comprising:

memory in which a directory table is stored, the directory table including an configurable cache state associated with the data block; and

a cache controller that speculatively updates the data block's cache state in the directory table upon receiving a memory request and before the apparatus receives a coherence response from the owner node.

15. (Previously presented) The apparatus of claim 14 wherein, upon receiving the coherence response from the owner node, the cache controller confirms the speculatively updated cache state if the updated cache state comports with the cache information provided in the coherence response.

16. (Previously presented) The apparatus of claim 14 wherein, upon receiving the coherence response from the owner node, the cache controller changes the speculatively updated cache state if the updated cache state does not comport with the cache information provided in the coherence response.

X. EVIDENCE APPENDIX

Not applicable.

XI. RELATED PROCEEDINGS APPENDIX

Not applicable.